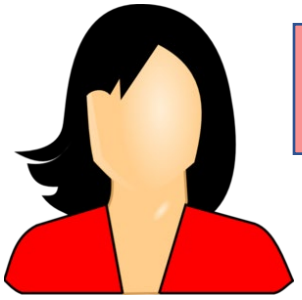
CS 4173/5173
COMPUTER SECURITY

Authentication Design

DIFFIE-HELLMAN: PROCESS

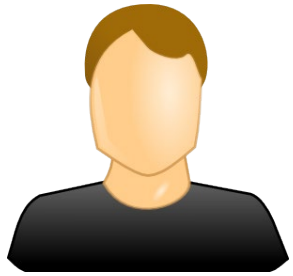


Alice

No public/private key

Public knowledge g, p

Bob



No public/private key

Generate random number S_A

Generate random number S_B

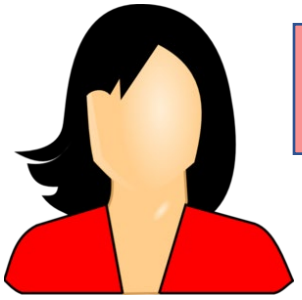
Compute and send $T_A = g^{S_A} \bmod p$

Compute and send $T_B = g^{S_B} \bmod p$

Compute $T_B^{S_A} \bmod p = \underline{\mathbf{g^{S_A S_B} \bmod p}}$

Compute $T_A^{S_B} \bmod p = \underline{\mathbf{g^{S_A S_B} \bmod p}}$

SECURITY ANALYSIS



Alice

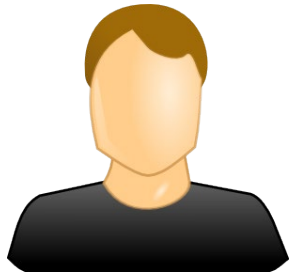
No public/private key

Public knowledge *g*, *p*

Generate random number S_A

secrets

Generate random number S_B



Bob

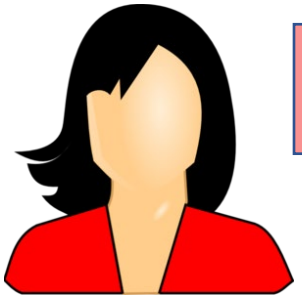
No public/private key

Compute and send $T_A = g^{S_A} \bmod p$

Can an attacker get S_A from T_A, why??

S_A is the discrete logarithm of g^{S_A} mod p

SECURITY ANALYSIS II

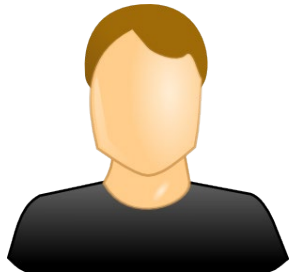


Alice

No public/private key

Public knowledge g, p

Bob



No public/private key

Generate random number S_A

Generate random number S_B

Compute and send $T_A = g^{S_A} \bmod p$

Compute and send $T_B = g^{S_B} \bmod p$

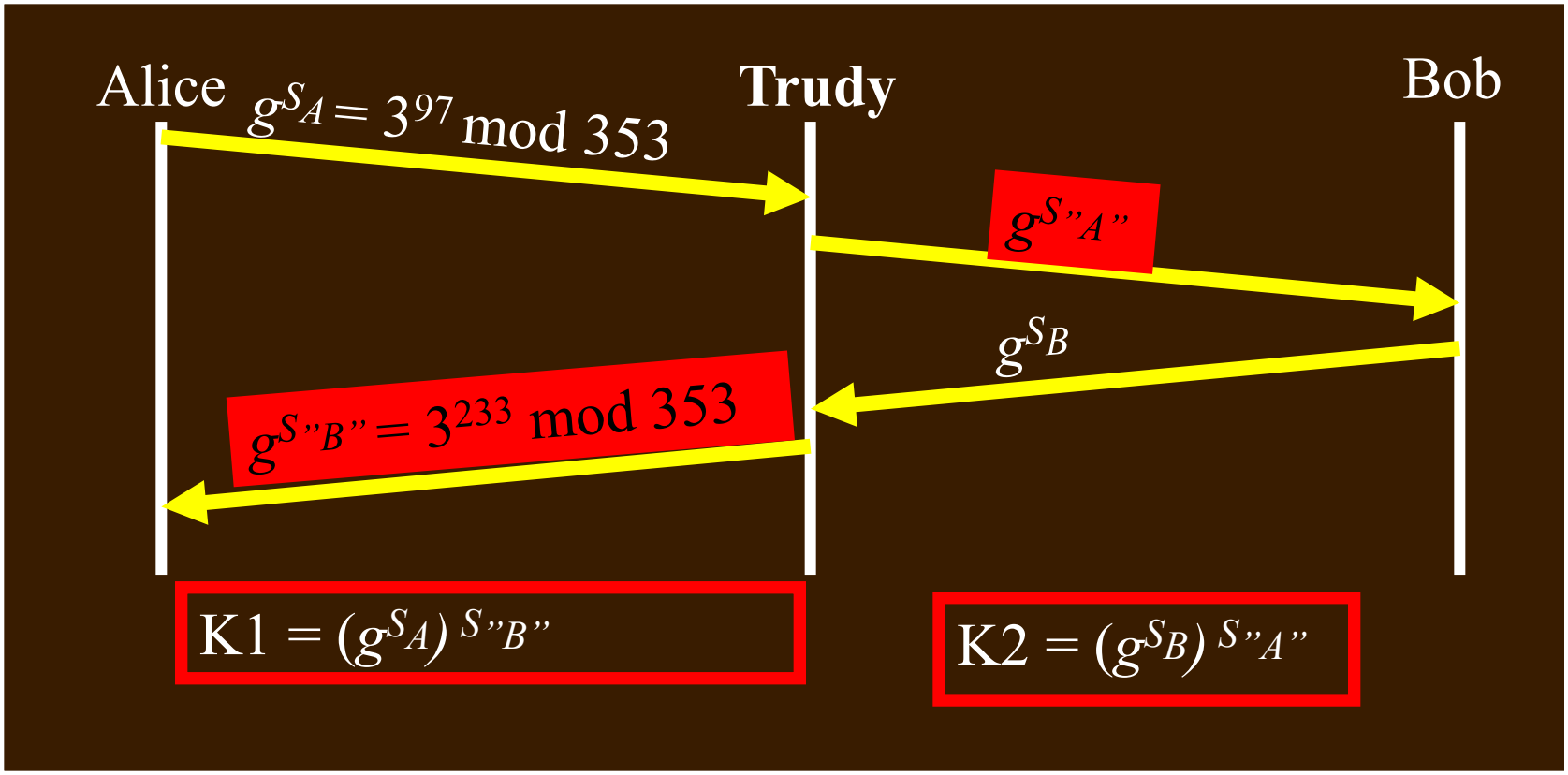
Compute $T_B^{S_A} \bmod p = g^{S_A S_B} \bmod p$

Compute $T_A^{S_B} \bmod p = g^{S_A S_B} \bmod p$

Can an attacker get $g^{S_A S_B} \bmod p$ from T_A and T_B?

MAN-IN-THE-MIDDLE ATTACK

- Trudy impersonates as Alice to Bob, and also impersonates as Bob to Alice



CERTIFICATES

- A CA is involved in authenticating users' public keys by generating certificates
- A certificate is a signed message vouching that a particular name goes with a particular public key
- Example:
  1. [Alice's public key is 876234]$_{carol}$
  2. [Carol's public key is 676554]$_{Ted}$ & [Alice's public key is 876234]$_{carol}$

- Knowing the CA's public key, users can verify the certificate and authenticate Alice's public key

EXAMPLE

- CA – everyone knows CA's public key.
  - CA is trusted.
- Alice wants to communicate to the real Bob
  - She sends a request to CA
  - Obtains a digital certificate from CA:

Bob's public key is
1902A12B2318871BF1
Expiration: 1/1/2020
[signed by CA]

Bob's D-H g, p, and T are
129381,102A7182019284FF, 910A81213
Expiration: 1/1/2020
[signed by CA]

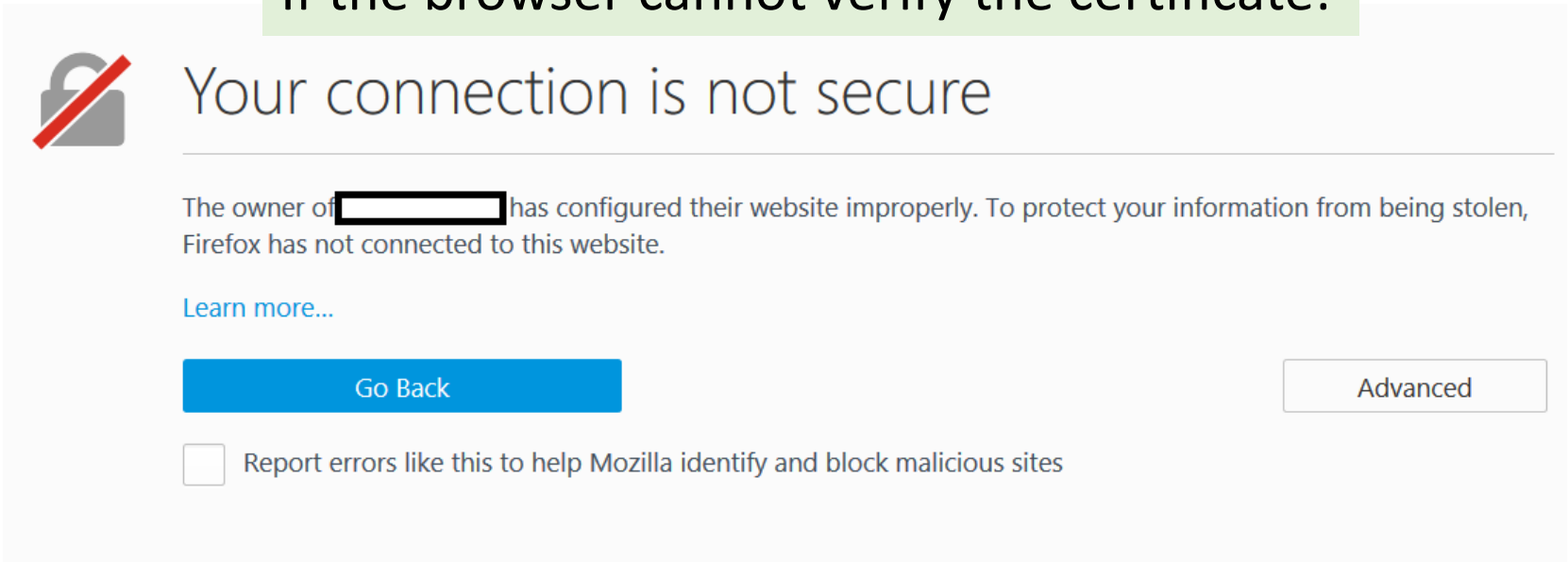
Q: digital certificate vs digital signature?

YAHOO'S CERTIFICATE

Certificate Hierarchy

- VeriSign Class 3 Public Primary Certification Authority - G5
  - Symantec Class 3 Secure Server CA - G4
    - www.yahoo.com

If the browser cannot verify the certificate:

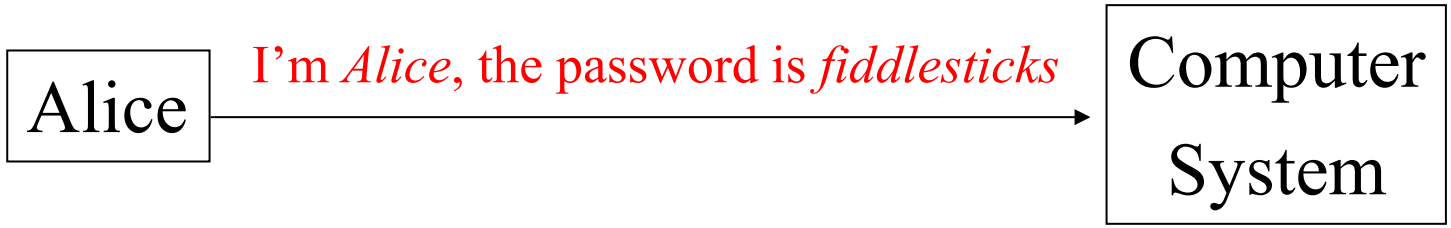


AUTHENTICATION

- Authentication is the process of reliably verifying certain information.
- Examples
  - Message authentication
    - Verify that a message has not been altered without proper authorization.
      - We have already learned: CBC-MAC, HMAC, RSA, …
  - User authentication
    - Allow a user to prove his/her identity to another entity (e.g., a system, a device).

AUTHENTICATION MECHANISMS

- Password-based authentication
  - Use a secret quantity (the password) that the prover states to prove he/she knows it.
  - Threat: password guessing/dictionary attack
    - a dictionary attack is to try a large number of possibilities of passwords.



AUTHENTICATION MECHANISMS (CONT'D)

- Address-based authentication
  - Assume the identity of the source can be inferred based on the network address from which packets arrive.
  - Threat: Spoof of network address
    - Not authentication of source addresses

AUTHENTICATION MECHANISMS (CONT'D)

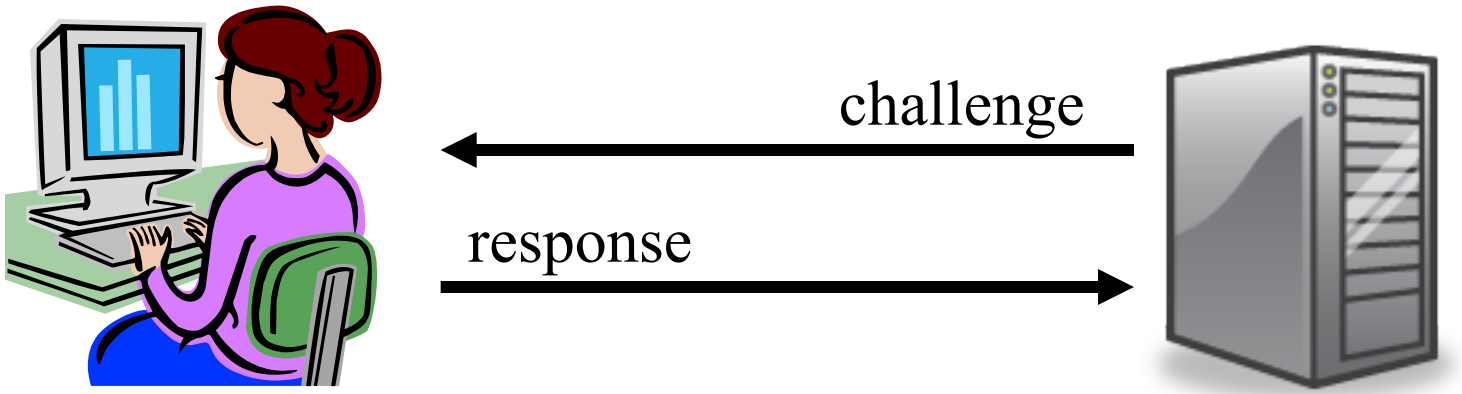
- Cryptographic authentication protocols
  - Basic idea:
    - A prover proves some information by performing a cryptographic operation on a quantity that the verifier supplies.
  - Usually reduced to the knowledge of a secret value
    - A symmetric key
    - The private key of a public/private key pair

CS 4173/5173
COMPUTER SECURITY

Password Authentication

PASSWORD-BASED USER AUTHENTICATION

- User demonstrates knowledge of a secret value to authenticate
  - most common method of user authentication



SOME ISSUES FOR PASSWORD SYSTEMS

- A password should be easy to remember but hard to guess
  - that's difficult to achieve!
- Some questions
  - what makes a good password?
  - where is the password stored, and in what form?
  - how is knowledge of the password verified?

PASSWORD STORAGE

- Storing unencrypted passwords in a file is high risk
  - compromising the file system compromises all the stored passwords
- Better idea: use the password to compute a one-way function (e.g., a hash, an encryption), and store the output of the one-way function
- When a user inputs the requested password…
  1. compute its one-way function
  2. compare with the stored value

ATTACKS ON PASSWORDS

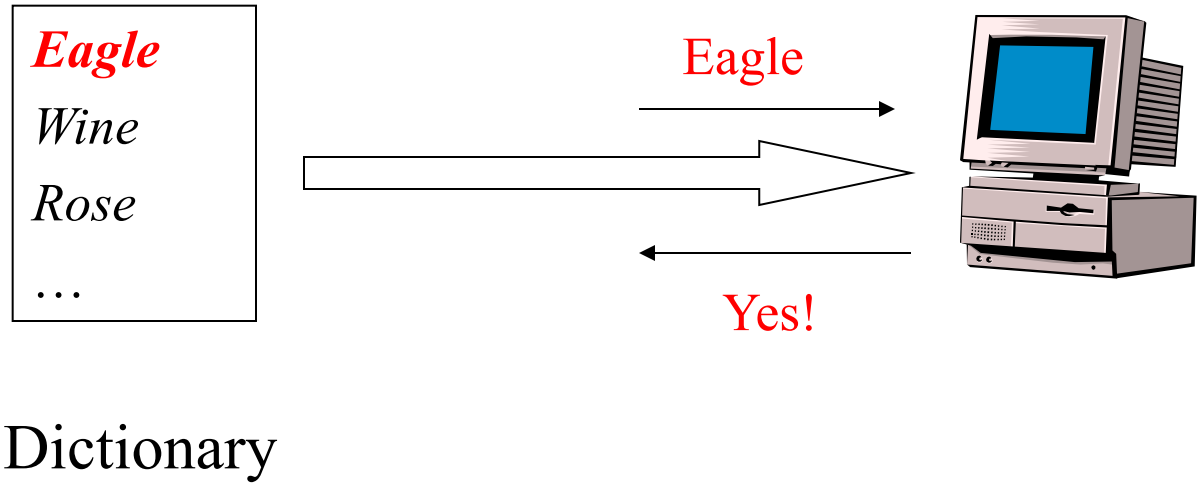
- Suppose passwords can be from 1 to 9 characters in length
- Possible choices for passwords = $26^1 + 26^2 + ... + 26^9 = 5 * 10^{12}$
- At the rate of 1 password per millisecond, it will take on the order of 150 years to test all passwords
  - Play with https://www.grc.com/haystack.htm
- Unfortunately, not all passwords are equally likely to be used

COMMON PASSWORD CHOICES

- Pet names
- Common names
- Common words
- Dates
- Variations of above (backwards, append a few digits, etc.)

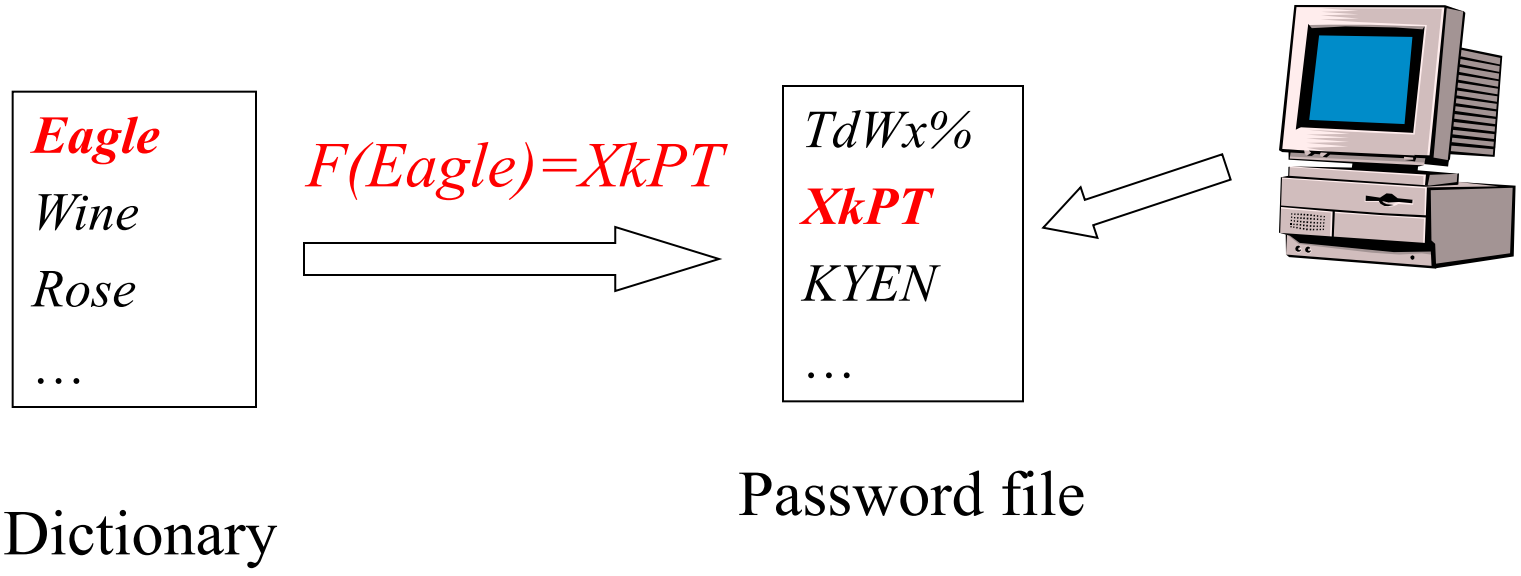
DICTIONARY ATTACKS

- Attack 1 (online):
  - Create a dictionary of common words and names and their simple transformations
  - Use these to guess the password



DICTIONARY ATTACKS (CONT'D)

- Attack 2 (offline):
  - Usually *F* is public and so is the password file
    - Most of the time, *F* is known hash function
  - Compute *F*(*word*) for each word in the dictionary
  - A match gives the password

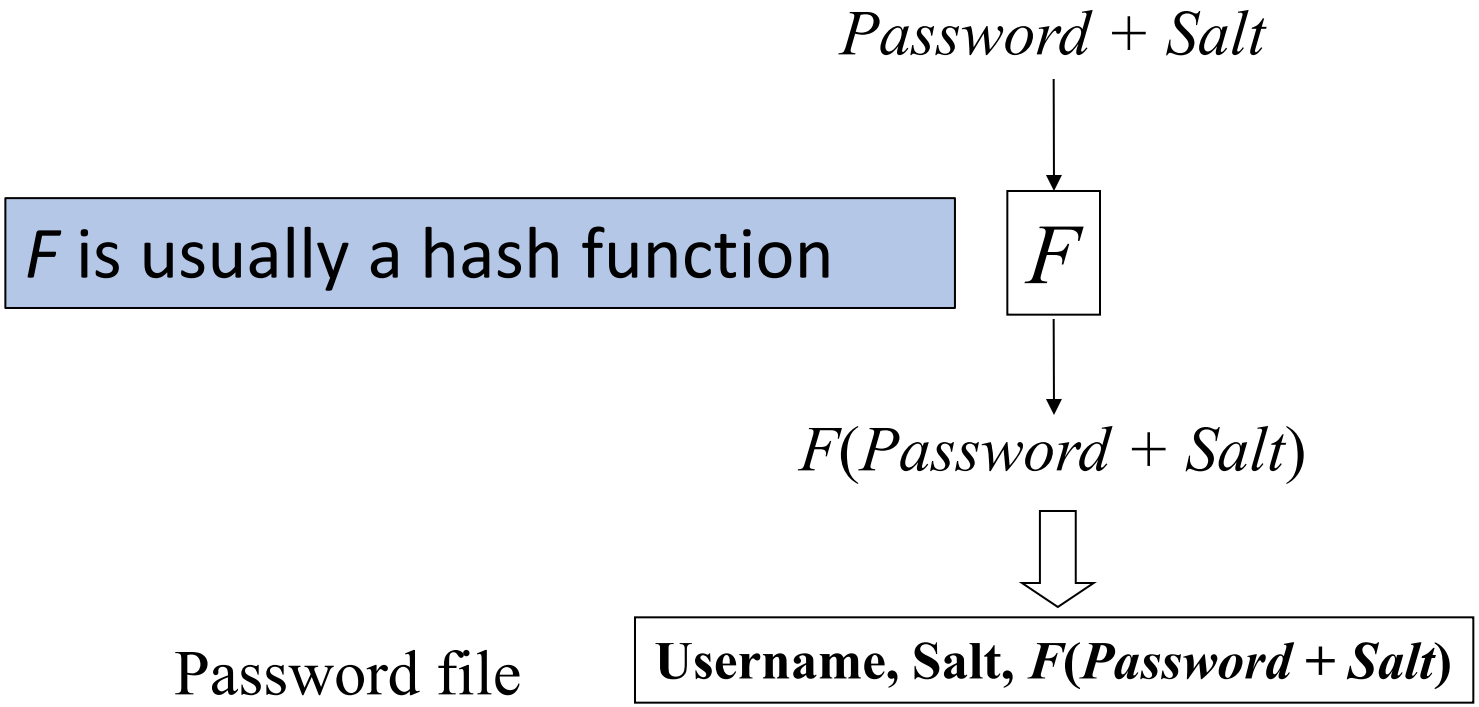


PASSWORD SALT

- To make the dictionary attack a bit more difficult
- Salt is a n-bit number between 0 and 2^n
- Derived from, for example, the system clock and the process identifier

PASSWORD SALT (CONT'D)

- Storing the passwords



Ref: https://www.cyberciti.biz/faq/understanding-etcshadow-file/

PASSWORD GUIDELINES FOR USERS

1. Initial passwords are system-generated, have to be changed by user on first login
2. User must change passwords periodically
3. Passwords vulnerable to a dictionary attack are rejected
4. User should not use same password on multiple sites

OU.EDU and OUHSC.EDU Staff, Faculty, Student, and Sponsored Accounts

All current and new accounts will be subject to new password requirements.

New Password Requirements:

- Minimum of 12 characters
- Must contain at least one upper AND lowercase letter
- Must contain at least one numeral OR one symbol
- Must be changed at least every 365 days
- Cannot be the same as your previous six passwords
- Cannot contain the words boomer, sooner, qwerty, or password
- Cannot contain your name or OUNet ID (ex. Jane Doe would not be able to use "jane", "doe" or doe0023)

For recommendations and guidelines, please see "How do I create a strong password?".

OTHER PASSWORD ATTACKS

- Technical
  - eavesdropping on traffic that may contain unencrypted passwords
  - "Trojan horse" password entry programs
- "Social"
  - careless password handling or sharing
  - phishing

CS 4173/5173
COMPUTER SECURITY

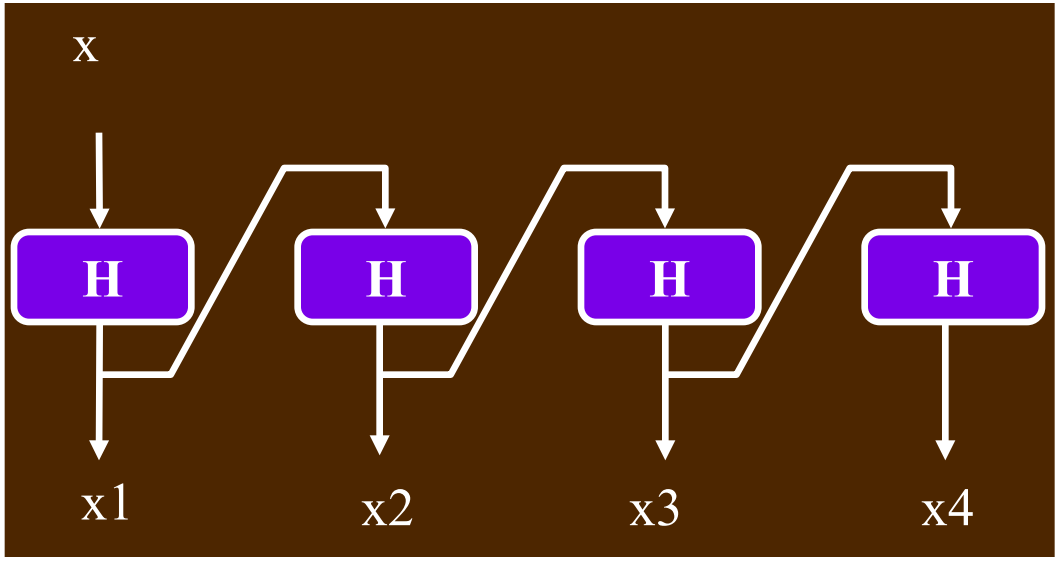
The S/Key Protocol

USING "DISPOSABLE" PASSWORDS

- Simple idea: generate a long list of passwords, use each only one time
  - attacker gains little/no advantage by eavesdropping on password protocol, or cracking one password
- Disadvantages
  - storage overhead
  - users would have to memorize lots of passwords!
- Alternative: the S/Key protocol
  - based on use of one-way (e.g. hash) function

S/KEY PASSWORD GENERATION

1. Alice selects a password **x**
2. Alice specifies *n*, the number of passwords to generate
3. Alice's computer then generates a sequence of passwords
   - $x_1 = H(\mathbf{x})$
   - $x_2 = H(x_1)$
   - …
   - $x_n = H(x_{n-1})$



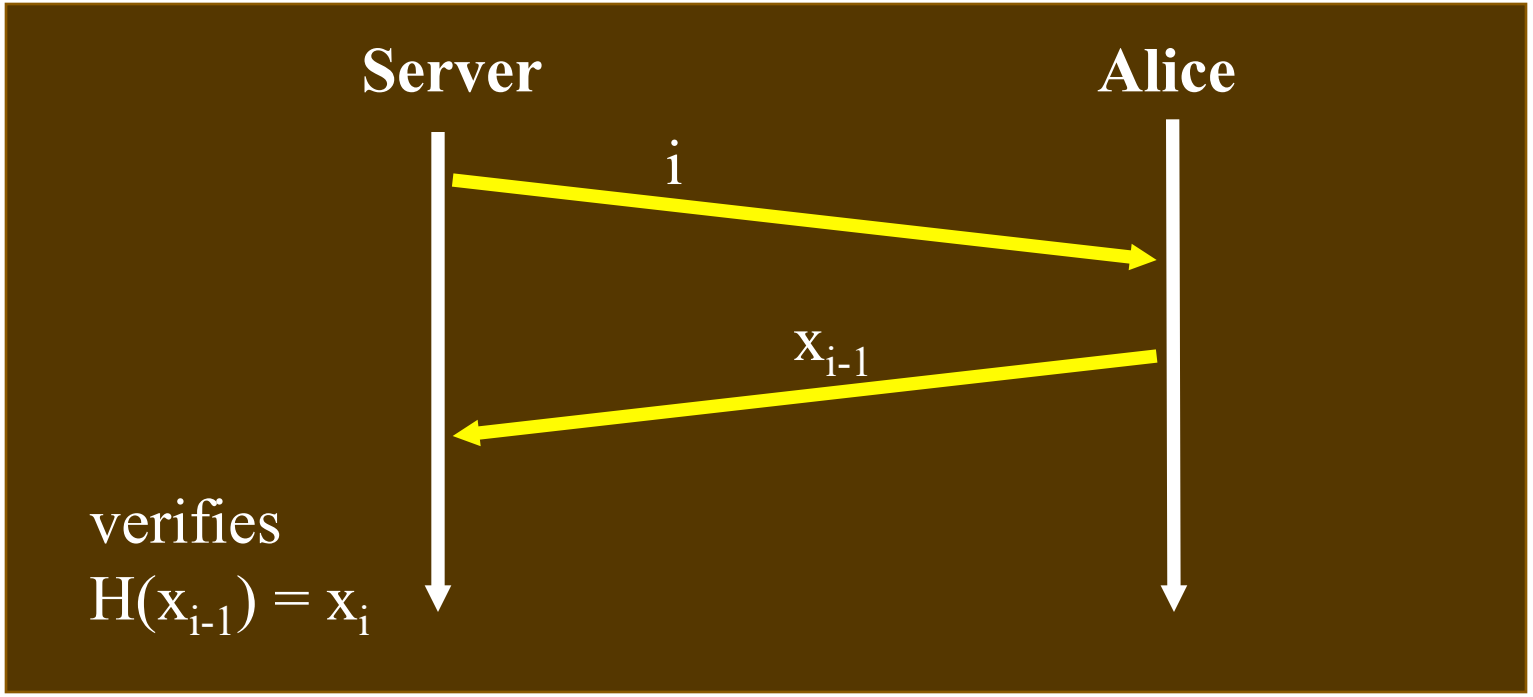
GENERATION… (CONT'D)

4. Alice communicates (securely) to a server the last value in the sequence: x_n

- Key feature: no one knowing x_i can easily find an x_{i-1} such that $H(x_{i-1}) = x_i$
  - only Alice possesses that information

AUTHENTICATION USING S/KEY

- Assuming server is in possession of x_i …



Is dictionary attack still possible?

LIMITATIONS

- Value of *n* limits number of passwords
  - need to periodically regenerate a new chain of passwords
- Does not authenticate server! Example attack:
  1. real server sends *i* to fake server, which is pretending to be Alice
  2. fake server sends *i* to Alice, who responds with x_{i-1}
  3. fake server then presents x_{i-1} to real server

BIOMETRICS

- Relies upon physical characteristics of people to authenticate them
- Desired properties
  1. uniquely identifying
  2. very difficult to forge / mimic
  3. highly accurate
  4. easy to scan or collect
  5. fast to measure / compare
  6. inexpensive to implement

ASSESSMENT

- Convenient for users (e.g., you always have your fingerprints, never have to remember them), but…
  - potentially troubling sacrifice of private information
  - no technique yet has all the desired properties

ASSESSMENT (CONT'D)

Biometrics	Univer-sality	Unique-ness	Perma-nence	Collect-ability	Perfor-mance	Accept-ability	Circum-vention
Face	H	L	M	H	L	H	L
Fingerprint	M	H	H	M	H	M	H
Hand Geometry	M	M	M	H	M	M	M
Keystroke Dynamics	L	L	L	M	L	M	M
Hand vein	M	M	M	M	M	M	H
Iris	H	H	H	M	H	L	H
Retina	H	H	M	L	H	L	H
Signature	L	L	L	H	L	H	L
Voice	M	L	L	M	L	H	L
Facial Thermogram	H	H	L	H	M	H	H
DNA	H	H	H	L	H	L	L

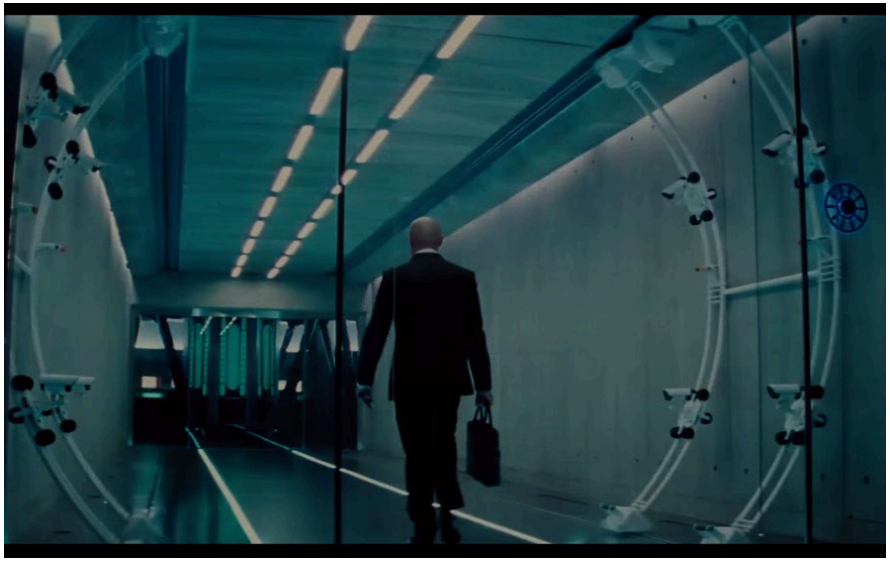
H=High, M=Medium, L=Low

EXAMPLE BIOMETRIC TECHNOLOGIES

- Signature / penmanship
- Fingerprints
- DNA
- Palm geometry
- Retina scan
- Iris scan
- Face recognition
- Voice recognition

BEHAVIOR AUTHENTICATION

- Human behavior depends on a person's habit, education, living environment, family, ….
- Data from computers/sensors reflects human behavior, and can be sometimes used to authenticate the identity of a person.



In Mission Impossible 5